Remarks

The above Amendments and these Remarks are in reply to the Office Action mailed 21 April 2003. No fee is due for the addition of any new claims. An appropriate Petition for Extension of Time to Respond is submitted herewith, together with the appropriate fee.

Claims 1-15 and 17-28 were pending in the Application prior to the outstanding Office Action. In the Office Action, the Examiner rejected all claims. The present Response cancels claims 1-15, amends claims 17, 26-28, and adds new claims 29-44, leaving for the Examiner's present consideration claims 17-44. Reconsideration of the rejections is requested.

I. DISCUSSION OF THE REJECTIONS

The Examiner is thanked for the courtesy of her telephone interview with the undersigned on July 2, 2003. During the telephone interview, claim 1 was discussed, and claim 17 was discussed briefly. Applicants have now canceled claims 1-15, and introduced new claims 29-44. The rejection of claims 1-15 is therefore considered moot. Nevertheless, Applicants will provide some comments on the relationship of the cited references to new independent claim 29. Thereafter, Applicants will discuss new dependent claims 30-41, independent claim 17, and dependent claims 18-25.

A. New Claim 29

During the discussion, it became clear that the Examiner was of the view that Lamping '632, and more specifically Lamping '250 as incorporated into Lamping '632, teaches Applicant's

previously claimed step of obtaining layout data indicating an element's position *relative to a parent* in the space with negative curvature. If this is true, however, then the representation of the position in relative terms is taught only as an *intermediate step* in the process of calculating the layout position data for each element.

Applicants will now discuss the teachings of Lamping '632, then the teachings of Lamping '250. Applicants will then point out how Applicants' new claim 29 is written to claim a situation in which the layout position data is represented in the data structure in relative terms *after it is fully calculated.*

1. <u>Lamping '632</u>

Lamping '632 does not teach that layout position data in the *final* data structure is represented only *relative* to a parent element. In order to identify the relevant parts of Lamping within which to look for such a teaching, it will be helpful to understand the typical basic flow of Lamping '632.

As set forth in Lamping '632, col. 16, lines 53-62 (among other places), the system first uses node-link data defining a node-link structure to obtain "<u>layout data</u>", indicating "positions" for parts of the node-link structure in a layout space with negative curvature, such as a hyperbolic plane. Then the system uses the layout data to obtain "<u>mapped data</u>" for a representation of the node-link structure on a display. See also Lamping '632, Fig. 8 steps 304-306, and corresponding text at col. 19, lines 7-10:

```
The act in box 304 uses node-link data 250 to obtain
layout data 252. The act in box 306 then uses layout
```

```
data 252 to map to a unit disk and to map the unit disk
to a circular region of display 208.
```

Applicants' new claim 29 (and prior claim 1) are (were) concerned with only the first step of this process, that of obtaining "layout data" indicating "positions" for parts of the node-link structure in a layout space with negative curvature. The sections of Lamping '632 that discuss using the layout data to obtain "mapped data" for display are not pertinent to Applicants' claim 29 because in Lamping '632, the mapped data is not in a space with negative curvature as called for in Applicants' claim.

The sections of Lamping '632 that discuss the step of obtaining "layout data" in the space with negative curvature are set forth primarily in Fig. 9 and the accompanying text at col. 20, line 18 through col. 21, line 8. The routine operates recursively, starting with the handle of a root node in the node-link data, with "coordinates of a position" for the root node in the hyperbolic plane, and with a pie-shaped region (wedge and room bound) within which the node's children will be laid out. (Lamping '632, col. 20, lines 20-28.) Thus before any children of the root node are laid out, the "position" of the root node is already known.

If the node currently being handled has children, then the routine calculates a "distance" from the current node to its children (step 354). The distance is one coordinate for each of the child nodes, but Lamping '632 does not indicate that a second coordinate, such as an angle, is known at this point for any of the children. It will be seen below that Lamping '250 reveals that the second coordinate is *not* known at this point. Thus at this point in the routine, the "position" is not yet *fully* known, either in relative or absolute terms, for any of the children of the root node.

The routine then begins an iterative loop that handles each of the current node's children in turn. Each iteration begins (in step 362) by obtaining, for the next child node, a "position in the hyperbolic plane", and a wedge and room bound within which the child node's further children will be laid out. Thus by the end of step 362, *all* coordinates of the current child's "position" are known. In addition, it can be seen that the "position" of each child is known before the routine loops around to begin handling the next sibling.

In step 364, the routine then makes a recursive call to lay out the current child's *further* children (if any), providing the handle of the current child and its "position", wedge, and room bound from step 362. Thus before the call is made to lay out any *further* children (step 364), the "position" of the current child and the *distance* to its further children are known. Again, however, Lamping '632 does not indicate that the *angle* to any of the further children are known at this point. That determination is left for the performance of step 364 in the context of handling each individual further child node.

Whenever the recursion reaches the depth of a leaf node (a node with no further children, step 352), the routine determines a "radius" for the node (step 370), and creates a "data structure" for the node (step 372). The routine creates such a data structure also for each parent node after data structures have already been created for all the children of the parent node (step 360). As pointed out at Lamping '632, col. 21, lines 2-3, the data structure for the node includes its "position", and its "radius" from step 370). (Note that the "radius" of a node is not part of its "position"; it is only an indication of how much space will be allocated to the node for visual features upon subsequent rendering.)

Nothing in any of the above excerpts from Lamping '632 states whether the node "positions" identified in these data structures are expressed in absolute or relative terms after completion of the layout routine. While one or more coordinates of the position may have been expressed in relative terms at some point as an *intermediate step* during the calculation, Lamping '632 does *not* teach that any node position is expressed in relative terms *as written into the final data structure*. Should the Examiner believe otherwise, it is respectfully submitted that in order to make a prima facie case of anticipation, it is up to the Examiner to point out where this is taught in Lamping '632.

2. Lamping '250

Nor does *Lamping '250*, referred to in Lamping '632 as the "Layout Application", set forth any example in which a node position is expressed in relative terms as written into the final data structure. Lamping '632 invokes Lamping '250 to describe a technique for obtaining a "distance" to the children in the hyperbolic plane, in step 354 (Lamping '632, col. 20, line 43). But step 354 is only an intermediate step in the developing of the position of the current node, to be stored in the final data layout structure. The distance to the children is used to *determine* each child node's "position", wedge and room bound (Lamping '632, col. 20, lines 46-50). The Lamping '250 example does not say that this distance is ever represented in the final Layout data structure.

Lamping '632 invokes Lamping '250 also to describe techniques for performing such determination of each child node's "position", wedge and room bound based on the "distance"

(Lamping '632, col. 20, lines 49-50). One might therefore look to Lamping '250 for any teaching that the "position" calculated by such techniques is relative or absolute. But Lamping '250, too, never teaches that any "position" calculated for a *final* data structure, be expressed in relative terms.

In Lamping '250, the layout process example described therein is described primarily with respect to Figs. 9 and 10 and the text at col. 21,line 33 - col. 24, line 46. Fig. 9 and the accompanying description at '250, col. 21, line 33 - col. 22, line 40, is very similar to the discussion in Lamping '632 of the step of obtaining layout data in the space with negative curvature. Lamping '250 then goes on, however, to discuss details of certain steps of his Fig. 9, including step 362, "Obtain Next Child's Position, Wedge, Room Bound", for a particular implementation.

The discussion refers to Fig. 10, which includes a step 400 of "...Obtain Child's Position, Wedge, Room Available". As it relates the obtaining of the child's "position", Lamping '250 begins his explanation of this step as follows:

> The act in box 400 can obtain preliminary width angle Wn as the product of the child's weight and the angle fraction, both obtained in box 390. The act in box 400 can use (θx, θy) and the previous width angle W0, either from box 398 or from the previous iteration, <u>to calculate child direction (θx, θy)</u> having coordinate values equal to (((θx (cos (W0 +Wn)))-(θy (sin (W0 +Wn)))), ((θy (cos (W0 +Wn)))+(θx (sin (W0 +Wn))))). Then the act in box 400 updates previous width W0 to have the value of the preliminary width Wn. (Lamping '250, col. 23, lines 26-34; emphasis added).

Thus the first step in calculating the child's "position" is to calculate its second coordinate, the *angle*. Clearly, therefore, the "position" of the current node in layout space was not fully known, in relative or absolute terms, prior to step 400 (i.e. prior to step 362 in Lamping '632).

Lamping '250 then continues:

> The act in box 400 can obtain the next child node's <u>position</u> using the <u>position (x, y)</u> of the parent node, from box 350 if the parent node is the root node or from box 364 when performed for the parent node; using the child direction (θx, θy); and using the <u>distance D</u> from box 394 or 396. The <u>position include values (x, y, θx, θy) that are calculated using transformations</u>. (Lamping '250, col. 23, lines 35-40; emphasis added).

> The act in box 400 can obtain a first translator from the origin to the position of the parent node, then <u>obtain the new x- and y-coordinates by applying the first translator to the x- and y-components of the distance D</u>, (Dθx, Dθy), where (θx, θy) is the child direction described above. The act in box 400 can then <u>obtain a second translator to the origin for the new x- and y- coordinates</u>, and apply the second translator to the results of applying the first translator to the child direction (θx, θy), to obtain the new values for ((θx, θy). (Lamping '250, col. 24, lines 4-10; emphasis added).

In other words, the "position" (x- and y- coordinates) of the current child node position is obtained by starting with the (x,y) "position" of the parent node in layout space, and then applying the distance D and direction (θx, θy) from the parent node to the child node, to obtain new x- and y- coordinates in layout space for the child node. This is a transformation which *converts* a relative position (Dθx, Dθy), which establishes the child node's position only *relative*

to the parent node's position, into an absolute position, which establishes the child node's position relative to the *origin*.

Thus by the end of step 400 in Lamping '250, the current node position is expressed in *absolute* terms, not *relative* to any other node position (except for children of the root node, for which the absolute and relative positions may be the same).

And this is the expression of the node position that is passed back to step 364 at the bottom of the flow chart of Fig. 10. Step 364 (Lamping '250, Fig. 9), it will be recalled, is the step that recursively calls the layout routine with the child node's handle, wedge, room bound, and "position".

Thus while the routine of Fig. 9 might represent certain node positions in relative terms within steps 354-362 of Fig. 9, by the time the layout routine of Fig. 9 is called recursively to lay out the next child node, node positions are expressed only in *absolute* terms.

Moreover, prior to step 400 in Fig. 9, the "position" of the current node is not fully developed, even in relative terms. Step 400 calculates the second coordinate of the node's position (the *angle*), then immediately uses the parent node's *absolute* position, together with the distance information and the newly determined angle information, to calculate the *absolute* position of the current node.

Accordingly, while Lamping '632 and '250 might teach obtaining layout data indicating an element's position *relative to a parent* in the space with negative curvature, it does so only as an *intermediate step* in the process of calculating the layout position data for each element. By the time the element's position is written into the node layout data structure, and before the

position of the next node is determined even in relative terms, the element's position is expressed only in *absolute* terms. In particular, there is never a time when the position of more than one node, whose position has been fully determined, is expressed relative to any other node in the node-link structure being laid out (except perhaps nodes that are children of the root node).

Again, should the Examiner believe otherwise, it is respectfully submitted that in order to make a prima facie case of anticipation, it is up to the Examiner to point out where this is taught in either Lamping '632 or Lamping '250.

3. New Claim 29 calls for the layout position data for *each* of a *plurality* of elements to be laid out, to be represented only in relative terms in the data structure after all elements in the plurality have been laid out.

New claim 29 calls for, among other things, the steps of:

A method of laying out a plurality of elements of a node-link structure in a space with negative curvature, the method comprising:

...

storing the positions for each element in the plurality in a data structure such that after the positions for all elements in the plurality have been calculated, the position of each element in the plurality is stored in the data structure only relative to an element of the node-link structure other than a root element of the node-link structure. (emphasis added.)

As can be seen, this claim calls for the layout process to identify node positions in relative terms, other than solely as an intermediate step in the calculation of one node's position, by claiming a method of laying out a *plurality* of elements of a node-link structure. The claim calls for the position of *each* element in the plurality to be stored in the data structure only relative to another element of the node-link structure (other than the root). In other words, the positions of

at least two elements of the node-link structure must be stored in the data structure only in relative terms.

This feature is not described in either Lamping patent. In Lamping '250, in fact, the description is quite clear that position data exists in relative form only as an *intermediate step* in the calculation of the "position" of each node. As soon as the position of a node is known in relative terms, it is immediately converted to absolute terms before the routine begins determining the position of the next node. Never, in Lamping '250, is the position in layout space of *more than one node* represented in the data structure only in relative terms.[1]

Nor would it be obvious to modify either Lamping patent to identify node positions in the space with negative curvature using relative coordinates, again other than as merely an intermediate step in the calculation of the position of each node. As set forth in detail in Applicants' specification at pp. 5-7, and in Applicants' Response C at pp. 13-14, this feature provides numerous advantages that are not even hinted at in either Lamping patent.

Accordingly, for these and other reasons, Applicants believe that new claim 29 should be patentable.

B. Dependent Claims 30-41

New claims 30-41 all depend ultimately from new independent claim 29. These claims should all be allowable because of their dependency from independent claim 29. These claims

---

[1]Note that the term "data structure" is used in new claim 29 in its general form. No regularity of structure is necessarily required. In addition, note that Applicants' discussion in this paper concerns the *teachings* of the earlier Lamping patents. No statement made herein should be considered as being relevant to the scope of any *claims* of the earlier Lamping patents.

also each add their own limitations which, it is submitted, render them patentable in their own right.

Several of the new dependent claims are similar to former claims 2-12. The Examiner rejected those claims 2-12 in the previous Office action as being anticipated by Lamping '632. In Response C, Applicants provided numerous points explaining why these claims should be patentable in their own right. The new Office action, however, does not appear to address any of Applicants' points. It merely repeats the prior rejection.

The new Office action does add a "Response to Amendment" section, but so far as Applicants can discern, this section is merely a sequence of quotations, and in some cases mis-quotations, from disparate locations in Lamping '632, taken out of context and cobbled together as a paragraph. The Examiner certainly intended something more significant by this section, but respectfully, Applicants cannot understand what it might be. The section does not identify any limitations in Applicants' claims, and does not make any correlation between such limitations and corresponding teachings in Lamping '632.

In light of this, Applicants submit that the Examiner has never made a prima facie case that any of the former claims 2-12 were unpatentable. In any event, however, the rejections of former claims 2-12 are now moot since they have been canceled.

C. Independent Claim 17

The Examiner rejected claim 17 as being anticipated by Lamping '632.

In Response C, Applicants pointed out that claim 17, among other things, emphasizes the ability of the methods described in the subject patent application to lay out only *parts* of the node-link structure in layout space, *without having to lay out the full tree, every time.* Thus the claim requires that the step of obtaining layout data identifying a subject element's position in layout space be based on only certain relationship data that *excludes relationships with at least one element of the node-link structure.*

As pointed out in Response C, Lamping '632 teaches layout methods that start with the root node. The Examiner has not cited anything in Lamping '632 that teaches a step of obtaining layout data identifying a subject element's position in layout space be based on only certain relationship data that *excludes relationships with at least one element of the node-link structure.*

In the new Office action, it appears that the Examiner did not address Applicants' points with respect to claim 17 at all. The Examiner still has not identified any teaching in Lamping '632 that teaches a step of obtaining layout data identifying a subject element's position in layout space be based on only certain relationship data that *excludes relationships with at least one element of the node-link structure.* If the Examiner responded to Applicants' points in the "Response to Amendment" section of the new Office action, Applicants respectfully cannot understand the response.

Applicants therefore repeat their assertion that the Examiner has not made a prima facie case of unpatentability.

In addition, claim 17 previously called for the obtaining of layout data identifying the subject element's position in the space with negative curvature, to be based on only the nearby

relationship data, and not on the position of any other element in the structure. Applicants have now amended claim 17 to emphasize that the layout data is to not to be based on the position of any other element in the structure. Applicants note that Lamping '632 and Lamping '250 both refer *extensively* to the positions of parent nodes in obtaining layout data identifying child nodes' positions in the space with negative curvature. Claim 17 distinguishes over such teachings by requiring that layout data identifying a subject element's position in the space with negative curvature be based on *only* the nearby relationship data, and *not on the position of any other element in the structure*.

Accordingly, it is respectfully submitted that the Examiner has not yet made a prima facie case that claim 17 should be rejected.

D. Dependent Claims 18-25

The Examiner rejected claims 18-25 in the previous Office action as being anticipated by Lamping '632. As with the rejections of dependent claims 2-12, in Response C, Applicants provided numerous points explaining why claims 18-25 should be patentable in their own right. Again, however, the new Office action does not appear to address any of Applicants' points. It merely repeats the prior rejection.

In light of this, Applicants once again submit that each of the dependent claims 18-25 should be patentable since the Examiner has not made a prima facie case of unpatentability.

E. Independent Claims 26-28

Claims 26-28 are independent claims all containing limitations similar to those in independent claim 17. These claims should all be patentable for many of the same reasons as set forth above with respect to claim 17. These claims also add their own limitations which, it is submitted, render them patentable in their own right. These claims have also been amended to provide the same additional emphasis described above with respect to claim 17.

It is respectfully submitted, therefore, that these claims should be patentable because the Examiner has not made a prima facie case of unpatentability.

II. CONCLUSION

In light of the above, it is respectfully submitted that all of the claims now pending in the subject patent application should be allowable because the Examiner has not made a prima facie case of unpatentability. A Notice of Allowance is therefore requested. The Examiner is respectfully requested to telephone the undersigned if he can assist in any way in expediting issuance of a patent.

Enclosed is REQUEST FOR CONTINUED EXAMINATION and an appropriate PETITION FOR EXTENSION OF TIME UNDER 37 C.F.R. § 1.136 for extending the time to respond up to and including October 21, 2003.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 50-0869 (Docket No. INXT 1002-1) for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: 10/21/2003

By: Warren S. Wolfeld

Warren S. Wolfeld, Reg. No. 31,454

Haynes Beffel & Wolfeld LLP
P.O. Box 366
Half Moon Bay, CA 94019
(650) 712-0340 phone